The Claims:

1. (Currently Amended) A method ~~for managing HPC node failure~~ comprising:

determining that one of a plurality of ~~HPC~~ nodes has failed, each ~~HPC~~ node comprising ~~an integrated fabric~~ a switching fabric integrated onto a board and one or more processors integrated onto the board; and

removing the failed node from a virtual list of ~~HPC~~ nodes, the virtual list comprising one logical entry for each of the plurality of ~~HPC~~ nodes.

2. (Currently Amended) The method of Claim 1, further comprising:

determining that at least a portion of ~~an HPC job~~ a job was being executed on the failed node; and

terminating at least the portion of the ~~HPC~~ job.

3. (Currently Amended) The method of Claim 2, further comprising:

determining that the ~~HPC~~ job was associated with a subset of the plurality of ~~HPC~~ nodes; and

deallocating the subset of ~~HPC~~ nodes from the job.

4. (Currently Amended) The method of Claim 3, each entry of the virtual list comprising a node status and the method further comprising changing the status of each of the subset of ~~HPC~~ nodes to ~~"available".~~ "available."

5. (Currently Amended) The method of Claim 3, further comprising:

determining dimensions of the terminated job based on one or more job parameters and an associated policy;

dynamically allocating a second subset of the plurality of ~~HPC~~ nodes to the terminated job based on the determined dimensions; and

executing the terminated job on the allocated second subset.

6. (Original) The method of Claim 5, the second subset comprising a substantially similar set of nodes to the first subset.

7. (Currently Amended) The method of Claim 5, wherein dynamically allocating the second subset comprises:

determining an optimum subset of nodes from a topology of unallocated ~~HPC~~ nodes; and

allocating the optimum subset.

8. (Currently Amended) The method of Claim 1, further comprising:

locating a replacement ~~HPC~~ node for the failed ~~HPC~~ node; and

updating the logical entry of the failed ~~HPC~~ node with information on the replacement ~~HPC~~ node.

9. (Currently Amended) The method of Claim 1, wherein determining one of the plurality of ~~HPC~~ nodes has failed comprises determining that a repeating communication has not been received from the failed node.

10. (Currently Amended) The method of Claim 1, wherein determining one of the plurality of ~~HPC~~ nodes has failed is accomplished through polling.

11. (Currently Amended) Software ~~for managing HPC node failure, the software~~ encoded in one or more computer-readable media and when executed operable to:

determine that one of a plurality of ~~HPC~~ nodes has failed, each node comprising ~~an integrated fabric~~ a switching fabric integrated onto a board and one or more processors integrated onto the board; and

remove the failed node from a virtual list of ~~HPC~~ nodes, the virtual list comprising one logical entry for each of the plurality of ~~HPC~~ nodes.

12. (Currently Amended) The software of Claim 11, further operable to:

determine that at least a portion of an ~~HPC~~ job was being executed on the failed node; and

terminate at least the portion of the ~~HPC~~ job.

13. (Currently Amended) The software of Claim 12, further operable to:

determine that the ~~HPC~~ job was associated with a subset of the plurality of ~~HPC~~ nodes; and

deallocate the subset of ~~HPC~~ nodes from the job.

14. (Currently Amended) The software of Claim 13, each entry of the virtual list comprising a node status and the software further operable to change the status of each of the subset of ~~HPC~~ nodes to ~~"available".~~ "available."

15. (Currently Amended) The software of Claim 13, further operable to:

determine dimensions of the terminated job based on one or more job parameters and an associated policy;

dynamically allocate a second subset of the plurality of ~~HPC~~ nodes to the terminated job based on the determined dimensions; and

execute the terminated job on the allocated second subset.

16. (Original) The software of Claim 15, the second subset comprising a substantially similar set of nodes to the first subset.

17. (Currently Amended) The software of Claim 15, wherein the software operable to dynamically allocate the second subset comprises software operable to:

determine an optimum subset of nodes from a topology of unallocated ~~HPC~~ nodes; and

allocate the optimum subset.

18. (Currently Amended) The software of Claim 11, further operable to:

locate a replacement ~~HPC~~ node for the failed ~~HPC~~ node; and

update the logical entry of the failed ~~HPC~~ node with information on the replacement ~~HPC~~ node.

19. (Currently Amended) The software of Claim 11, wherein the software operable to determine one of the plurality of ~~HPC~~ nodes has failed comprises software operable to determine that a repeating communication has not been received from the failed node.

20. (Currently Amended) The software of Claim 11, wherein the software operable to determine one of the plurality of ~~HPC~~ nodes has failed is accomplished through polling.

21. (Currently Amended) A system ~~for managing HPC node failure~~ comprising:

a plurality of ~~HPC~~ nodes, each node ~~including an integrated fabric~~ comprising a switching fabric integrated onto a board and one or more processors integrated onto the board; and

a management node operable to:

determine that one of the plurality of ~~HPC~~ nodes has failed, each node comprising an integrated fabric; and

remove the failed node from a virtual list of ~~HPC~~ nodes, the virtual list comprising one logical entry for each of the plurality of ~~HPC~~ nodes.

22. (Currently Amended) The system of Claim 21, the management node further operable to:

determine that at least a portion of an ~~HPC~~ job was being executed on the failed node; and

terminate at least the portion of the ~~HPC~~ job.

23. (Currently Amended) The system of Claim 22, the management node further operable to:

determine that the ~~HPC~~ job was associated with a subset of the plurality of ~~HPC~~ nodes; and

deallocate the subset of ~~HPC~~ nodes from the job.

24. (Currently Amended) The system of Claim 23, each entry of the virtual list comprising a node status and the management node further operable to change the status of each of the subset of ~~HPC~~ nodes to ~~"available".~~ "available."

25. (Currently Amended) The system of Claim 23, the management node further operable to:

determine dimensions of the terminated job based on one or more job parameters and an associated policy;

dynamically allocate a second subset of the plurality of ~~HPC~~ nodes to the terminated job based on the determined dimensions; and

execute the terminated job on the allocated second subset.

26. (Original) The system of Claim 25, the second subset comprising a substantially similar set of nodes to the first subset.

27. (Currently Amended) The system of Claim 25, wherein the management node operable to dynamically allocate the second subset comprises the management node operable to:

determine an optimum subset of nodes from a topology of unallocated ~~HPC~~ nodes; and

allocate the optimum subset.

28. (Currently Amended) The system of Claim 21, the management node further operable to:

locate a replacement ~~HPC~~ node for the failed ~~HPC~~ node; and

update the logical entry of the failed ~~HPC~~ node with information on the replacement ~~HPC~~ node.

29. (Currently Amended) The system of Claim 21, wherein the management node operable to determine one of the plurality of ~~HPC~~ nodes has failed comprises the management node operable to determine that a repeating communication has not been received from the failed node.

30. (Currently Amended) The system of Claim 21, wherein the management node operable to determine one of the plurality of ~~HPC~~ nodes has failed is accomplished through polling.